REMARKS

Reconsideration and allowance of the above-referenced application are respectfully requested. Claims 1-26 are unchanged, claims 27-33 are canceled without prejudice or disclaimer to the underlying subject matter, new claims 34-35 are added, and claims 1-26 and 34-35 are pending in the application.

The cancellation of claims 27-33 renders moot the rejection under 35 USC §101: Applicant reserves the right to continue prosecution of the claims 27-33 in a continuing application.

Claims 1-33 were rejected under 35 USC §103 in view of U.S. Patent Publication No. 2002/0023147 by Kovacs in view of U.S. Patent Publication 2004/0210635 by Raniere and US Patent No. 5,926,177 to Hatanaka et al. This rejection is respectfully traversed.

Each of the independent claims 1, 7, 13, 20, (canceled claim 27), and 35 specify that a service object (e.g., a model object, a view object, or a controller object) associated with a network service is ***transferred*** to a network node, such as a user interface device. For example, claim 1 specifies that the network-enabled user interface device includes a network interface device configured for "***receiving a corresponding one of the service objects via the open protocol network***", and a controller configured for "***executing the received one service object*** for providing the first network service to the user"; claim 7 (and new claim 35) specifies "***receiving, via an open protocol network, at least one service object*** ... for a first network service", and "***executing the at least one service object for the first network service***"; Claims 13 (and canceled claim 27) specifies "***supplying to the network-enabled user interface device via the open protocol network at least one of the service objects***", and claim 20 specifies "***supplying a first network service to a network-enabled user interface device*** ... including ***supplying at least one of the service objects via the open protocol network....***"

Hence, each of the independent claims specify transfer of a service object (one of a model object, a view object, or a controller object) to the network-enabled user interface device in order to provide the corresponding network service to the user interface device.

As described in further detail below, the applied prior art assume that the "***service***

objects" (e.g., model object, view object, and controller object) for a given network service are executed in a single device such as a server (see, e.g., Figs. 2-5 of Kovacs et al, Fig. 1 of Hatanaka et al.; page 9, lines 22-24 of the subject specification).

In contrast, the claims specify the ***transfer*** of a service object to the network-enabled user-interface device for execution of the transferred service object ***by*** the network-enabled user interface device in order to provide the corresponding network service.

Further, the claimed interaction exchange of ***service transaction messages*** between the model object view object, and controller object enables the claimed transfer of the service object, since each of the service objects can be independently controlled based on the exchange of the service transaction messages (see, e.g., page 6, lines 22-25 of the specification); as described in the dependent claims, the service transaction messages enable received object (e.g., the controller object) to be executed by the user interface device, whereas the model object and view object can be executed remotely (see, e.g., claims 2-5, 8-11, 16-18, 22-25).

Further, the service transaction messages enable any one of the messages to be transferred between network nodes ***during execution of the corresponding network service***, while maintaining a user-perceived continuous service of the service (see, e.g., claims 13 and 20 and page 10, lines 2-7 of the specification).

These and other features are neither disclosed nor suggested in the applied prior art.

As admitted in the Official Action, Kovacs et al. fails to disclose or suggest the "movement of service objects through a network." Moreover, Kovacs et al. provides no disclosure or suggestion of ***executing*** a service object (e.g., a model object, a view object, or a controller object) in a ***user interface device***, as claimed. Rather, Kovacs et al. consistently teaches that a client browser 11 contains none of the service objects.

As illustrated in Figs. 2-3, a given network service is implemented in a ***server*** based on a controller object 13 implemented as a Java **servlet** ("corresponding to a small program running on a **server**", para. 39), a model 14, and a view object implemented as a Java **Server** Page (JSP) (para. 44, lines 8-12, para. 48).

Further, Fig. 4 illustrates that a given service have may multiple views 12, 12' and 12" and that the browser may interact with different services (and that different controller objects may communicate with each other).

As apparent from the foregoing, however, Kovacs et al. provides no disclosure or suggestion whatsoever of the ***user device*** executing *any* service object, but rather teaches away from the claims by teaching that each of the service objects (model-view-controller) for a given network service should reside in the **server.**

Further, Kovacs et al. provides no disclosure or suggestion of any interaction exchange of ***service transaction messages,*** as claimed.

Applicant traverses the Examiner's assertion that Raniere and Kovacs et al. are "from the same field of endeavor, namely the **manipulation of a MVC paradigm for gathering information**." Raniere has no relation whatsoever to a model-view-controller paradigm, and fails to provide any reference at all to a model-view-controller based architecture, let alone any reference to a "model object" or a "model" as used to describe a model-view-controller architecture.

Further, Raniere et al. provides no reference whatsoever to any object other than a "data object", which is described in para. 34 as "any of several information types represented in any of a multiplicity of data formats", for example **images, video, text, spreadsheets, word processing files**, etc. Hence, the disclosed "data object" is not equivalent to the model, view, or controller objects as described in the claims, the subject application, Kovacs et al., or Hatanaka et al.

Raniere et al. is directed to sharing common data objects such as images, slide presentations, etc. with multiple users during a voice telephone conference, and is not within the field of the inventors' endeavor, namely providing network enabled telephony devices providing network services according to model-view-controller architectures; further, Raniere et al. is not reasonably pertinent to the particular problem with which the inventors were involved, namely providing dynamic transfer of service objects in a model-view-architecture. Raniere et al. provides no disclosure or suggestion of transferring ***service objects*** (e.g., model objects, view objects, or controller objects) for execution of the transferred service object to provide a network service to a client device, and as such is non-analogous art. *In re Wood*, 202 USPQ 171, 174

(CCPA 1979). *In re Oetiker*, 24 USPQ2d 1443, 1445 (Fed. Cir. 1992).

As described above, Hatanaka et al. also assumes that the model object, view object, and controller object are "static" and do not move between nodes. Hatanaka et al. teaches a "ViewProxy" that appears to the Model object and Controller object as the same View object, but which manages a "collection of views", where a user can switch between the views using a windowing system to show active windows and hide inactive windows (see col. 2, lines 50-58, Abstract). Hence, any change in an object state causes the ViewProxy (which is a single view object relative to the model object and controller object) to notify all views, although only the active views of the windowing system will display the changes (see, e.g., col. 4, lines 4-18).

Further, Hatanaka et al. provides no disclosure whatsoever for the claimed selective termination of the received one service object providing the first network service based on reception, via the open protocol network, of a second service object for ***a corresponding second network service***. Rather, Hatanaka et al. simply teaches at column 2, lines 7-10 that old display data is closed as new graphical information is presented. Further, Hatanaka et al. teaches away from the claimed invention by describing at col. 2, lines 11-25 that changing the view object for the same network service is to be avoided: as described on page 3, lines 15-18 of the specification, use of the model view controller architecture typically has been limited to instances where GUI interfaces may be utilized for different hardware configurations.

In other words, Hatanaka et al. simply describes at col. 2 replacing one view object (providing one view) with a new view object (providing another view) for the **same service**. The View Proxy also serves as a collection of views for the same service.

Hence, Hatanaka et al. does not disclose or suggest the selective termination based on ***reception, vvia the open protocol network, of a second service object*** for a corresponding ***second network service***.

Hence, the Official Action fails to establish a prima facie case of obviousness because none of the references teach or suggest the features as asserted in the rejection.

Further, the hypothetical combination neither discloses nor suggests the claimed transfer of ***service objects***, as claimed: Kovacs et al. and Hatanaka et al. assume the model, view, and controller objects reside in the same server, and the "data objects" of Ranier et al. cannot be considered a teaching of moving ***service** objects*, as claimed.

In view of the above, it is believed this application is in condition for allowance, and such a Notice is respectfully solicited.

To the extent necessary, Applicant petitions for an extension of time under 37 C.F.R. 1.136. Please charge any shortage in fees due in connection with the filing of this paper, including any missing or insufficient fees under 37 C.F.R. 1.17(a), to Deposit Account No. 50-1130, under Order No. 95-468, and please credit any excess fees to such deposit account.

Respectfully submitted,

Leon R. Turkevich
Registration No. 34,035

Customer No. 23164
Date: June 8, 2006